Specification #0: Fast Data Reduction

Version: 0.1 16/SEP/09

Changes

0.1: First iteration of a proper use case for the reimplementation of the fast data processing. This is based on the currently fast_dp implementation currently in use and the derived set of requirements.

Purpose

To go from one or more sweeps of X-ray diffraction data from a single crystal to a reduced intensity data file (e.g. MTZ format) as quickly as possible while still giving reasonably well reduced data. This should work with data from the detectors used at Diamond (e.g. ADSC, Mar CCD detectors and Pilatus 6M) with both the native formats and CBF equivalent, and data measured in more than one sweep (e.g. multi-pass or EDNA-strategy data collection.)

A primary aim of this is to give back:

- Refined cell constants
- An idea of the data quality
- Measurements for downstream processing e.g. difference map calculation

The seconary uses for this are:

- Provide beamline scienists / local contacts with a real-time idea of how the beamline is behaving
- Check that all of the images are ok (i.e. not blank) for example from the output of COLSPOT, NSTRONG pixels.

Preconditions

The following preconditions are necessary:

- The full data set is available
- A large amount of computer horsepower is available
- The contents of the image headers are accurate, e.g. distance, wavelength and beam centre.

The following preconditions are optional:

- The unit cell constants and symmetry are known

Postconditions

After processing the cell constants and pointgroup used for data reduction will be known, along with the overall summary statistics (e.g. what you get in the tail of the Scala log file.) The measurements will also be available as scaled and merged intensities with anomalous pairs separated in MTZ format. If the cell constants and spacegroup were given the data will be reindexed to give the closest match, otherwise the standard setting will be used for the most likely pointgroup.

Error States

The following error states may be defined:

- Data reduction failed, for computational reasons or due to data quality
- Data reduction was successful, but the unit cell and spacegroup given are not possible
- Images missing from the given sequence
- Images present but blank

Data Reduction Failed

This is the circumstance where something went wrong. Need to enumerate these. In some cases it will be that the data reduction jobs fell off the cluster.

Disagreement with Unit Cell / Spacegroup

If the unit cell and spacegroup were specified and they did not match one of the options from IDXREF then the user should be informed, however the cell constants and symmetry should be ignored and data reduction continued. This may occur when a new crystal form is found, so it is potentially valuable.

Images Missing

If there are apparently images missing (i.e. 1-65, 67-90 are visible) then an error should be raised and the offending images identified.

Images Blank

If there were untrapped problems during data collection it may be the case that there are blank images. These will be found by running XDS COLSPOT on all available frames (over the cluster again, perhaps) and the number of strong pixels reported on each frame identified. If there is a sudden change in this (e.g. one or a few blank images somewhere) this should be raised as an exception.

Process

The first pass implementation of this procedure was:

- Autoindex data in P1 with a subset of the images
- Using forkintegrate, integrate all data in a number of small batches
- Allow XDS CORRECT to decide on the correct pointgroup to use for the data
- Transform the unmerged data file from XDS CORRECT to mtz format using POINTLESS

- Merge the reflections with SCALA to get the merging statistics and a merged MTZ file, using "scales constant".

The following additional steps have been found to be necessary:

- Use POINTLESS and XDS CORRECT to decide on the correct pointgroup
- Allow user to specify the unit cell and symmetry
- Be more robust against failure
- Be more flexible.

On the latter, will move towards the use of templates for different XDS steps which may be kept in a library of their own. This will allow additional detectors to be defined and used easily.

License

During development this will not be distributed. When it is "finished" the resulting pipeline should be made available to CCP4 using an EDNA framework to provide the interface.